

GROUP ART 2172
AMENDMENT AFTER FINAL Patent

REMARKS

This application was filed on 05 October 2001 with 58 claims. In the first examination of the application dated 17 December 2003, the Examiner, objected to the Drawings, the Specification, and claim 10 because of a typographical error. The Examiner further rejected claims 1-58 under 35 U.S.C. §103(a) over U.S. Patent No. 6,088,694 entitled *CONTINUOUS AVAILABILITY AND EFFICIENT BACKUP FOR EXTERNALLY REFERENCED OBJECTS* to Burns et al. (Burns '694) in view of U.S. Patent No. 6,581,075 entitled *SYSTEM AND METHOD FOR DATABASE SYNCHRONIZATION* to Guturu et al. (Guturu '075). Applicant responded on 17 March 2004 and amended the abstract, claims 1, 4, 10, 20, 23, 24, 38, 39, 42, 43, 45, 56, 57, and 58; cancelling claims 12, 13, 31, 32, 49, and 50. Applicant requested to be allowed to submit corrections to the drawings when Attorney for Applicant receives copies of the drawings that were submitted and reviewed by the Examiner and the Official Draftsman.

The Examiner responded with a final rejection of claims under 35 U.S.C. §103(a) under Burns '694 and Guturu '075, as well as rejecting claims 1, 20, 39, 57 and 58 under 35 U.S.C. §112, first paragraph. In response, Applicant submit an Amendment After Final Rejection under 37 CFR 1.116 putting the claims in condition for allowance and/or better condition for appeal. In separate correspondence enclosed Applicant submits a Proposed Amendment to the Drawing. Applicant requests the Examiner to enter the amendments because they do not raise new issues, nor do they require a new search; the amendments incorporate limitations of preexisting and already searched dependent claims into the independent claims.

The Examiner responded with telephone call on 17 September 2004 indicating that he would allow claims 1-11, 14-18, 20-30, 33-37, 39-48, 51-55 and requested that he cancel claims 57-58 by Examiner's Amendment. Attorney for Applicants complied, requesting claims 57 and 58 be canceled with traverse so that a divisional application could be filed on the remaining claims. The Examiner mailed a Notice of Allowability on 24 September 2004 indicating that claims 1-11, 14-18, 20-30, 33-37,

39-48 and 51-55 are allowed. Then, Attorney for Applicants received an Advisory Action mailed 20 September 2004 indicating that the proposed amendments dated 28 July 2004 were not entered because they raise new issues that would require further consideration. Claims 1-11, 14-18, 20-30, 33-37, 39-48 and 51-55, however, would be allowed if the amendments would be submitted in a separate, timely filed amendment canceling the non-allowable claims.. This amendment follows in which claims 1, 20, 39, 52 are amended and claims 12, 13, 19, 31, 32, 38, 49, 50, 56-58 are canceled.

The Rejection of claims 1, 20, 39, 57 and 58 under 35 U.S.C. §112, first ¶

The Examiner rejected claims 1, 20, 39, 57 and 58 asserting that they fail to comply with written description requirement. The Examiner asserts that the specification, as originally filed, fails to provide support for “said object capable of being edited independently of said related metadata.” Applicant traverses this rejection.

Applicant maintains that a “loose transaction model” **is one** where the file can be edited independently of the metadata. It is well known that Applicant is allowed to be her/his own lexicographer. In the originally filed specification on pages 1, line 26 through page 2, line 11, it is stated that:

If file and meta-data updates are tightly coupled (i.e. both updates happen within a single unified transaction), a transaction coordinator typically ensures On the other hand, in systems where a loose transaction model is provided, and direct content edits are allowed, consistency between file-data and meta-data may not be guaranteed at all times. A need therefore clearly exists for an improved technique for providing a consistent view of file data and meta-data in the presence of a loose-transaction model.

Thus, if a “tightly couple transaction is one in which updates to the file and meta-data happen within a single unified transaction,” it follows that a “loose transaction model” is one where direct content edits to the file are allowed separately from direct content edits to the meta-data; otherwise, why would guaranteeing consistency between file-data and meta-data be a concern?

Further support for inclusion of “maintaining (or ensuring) meta-data and object-data consistency **in a loose transaction model** of object and meta-data updates” is given in the originally filed specification on page 8, line 31 through page 9, line 1. Additionally, the Examiner is directed to the originally filed specification at page 11, lines 21-26 which states, “[t]he file is updated using normal file system application program interfaces. The file can be updated via SQL Mediated Object Manipulation where a handle is obtained from the mediator, that is preferably a filename with an encrypted access token string embedded as part of the filename, and is supplied as the filename to the filesystem API. **The file may be updated several times before the file’s meta-data is updated in the database.**” (Emphasis added; abbreviations and reference numerals omitted). And again, on page 15, line 18-20, “[t]his covers the case where an updater has modified and then closed the file after releasing any file locks, but has not updated the corresponding meta-data.” On page 28, lines 11-26, “[a]s the foregoing embodiments of the invention illustrate, a loose transaction model for updates to a file and its corresponding meta-data through a mediator is useful for directly performing **in-place edits of content data residing on stores external to the indexed meta-data store** (the latter could be a DMBS). This is subject to the requirement of ensuring consistency between the file content and the associated meta-data from a reader’s perspective.” (Emphasis added)

Thus, Applicant requests the Examiner to remove the rejection of claims 1, 20, and 39 under 35 U.S.C. §112, first paragraph because the specification does provide for editing file content separately from editing the metadata relevant to that file. Besides, Applicant has removed the objectionable language “loose transaction model” from the rejected claims, reserving the right reinstate the language. Applicant has canceled the remaining rejected claims 57, 58.

The Rejection of claims 1-11, 14-30, 33-48, 51-58 under 35 U.S.C. §103(a)

The Examiner rejected claims 1-11, 14-30, 33-48 and 51-58 under 35 U.S.C. §103(a) over Burns '694 in view Guturu '075. In response, Applicant amends independent claims 1, 20, and 39 to distinctly point out and particularly claim that the "object can be edited independently of the related metadata."

In amending the claims, Applicant does not add new matter. Support for the **updating said externally stored object while said object is accessed without modifying said metadata in said database** in the originally-filed application is set forth as stated above in the remarks directed to the rejection under 35 U.S.C. §112, first paragraph and again on page 9, line 11 which states that "the file can be edited independently of the metadata."

The Examiner alleges that Burns '694 discloses the elements of the claimed invention, i.e., "a method of maintaining consistency of content of an object and metadata related," "storing said related meta-data and a reference to said object in a table," "the object being stored externally to said database," "said reference used to obtain a handle for directly accessing or manipulating said external object", and "obtaining a version number." The Examiner, however, admits that Burns does not explicitly disclose the steps of comparing the embedded version number with a version number of a latest committed version of an externally stored object to determine if the handle refers to a current version of the externally stored object.

Respectfully, Applicant maintains that Burns '694 does not teach or suggest the amended claim limitation that the "externally stored object can be edited without modifying the metadata in the database." Recall that this was the problem solved by the Applicant and one not addressed by either Burns '694 nor Guturu '075. Burns '694 at column 11, lines 30-44 explains that in carrying out the file append operations, fields in the file metadata are set, the file is linked and then is allowed to be opened and then appended. "In the DataLinks system, the DLFF module permits the append operations to take place by setting the appropriate field values in the

metadata [i.e., by editing the metadata]." (column 11, lines 56-59) Then the file and the metadata are updated in a transactional manner meaning the update to both are either performed completely or the update fails. (column 12, lines 4-5). Similarly, in the file update operations, fields in the file metadata are set [i.e., the metadata are edited] (column 12, lines 43-44), then the file is opened and update operations are performed by setting the appropriate field values in the metadata." (column 12, lines 48-50). Burns '694, moreover, provides for "transaction atomicity" (column 10, lines 65-66) when modifying the object, i.e., the appends and updates of both the file/object and the metadata are checked-in and conducted in a "transactional manner" (Burns '694 at column 12, lines 3-8 and column 13, lines 8-13) by updating a column in the database. (column 13, lines 5-15).

In fact, Burns '694 is described by Applicant in the originally-filed specification on page 1, lines 29-32 wherein the "file and meta-data updates are tightly coupled (i.e. both updates happen within a single unified transaction), a transaction coordinator typically ensures a consistent view by **locking out readers of meta-data** as well as file data until the transaction is committed. Intermediate/uncommitted updates to either are not visible." See Burns '694 at column 9, lines 8-18 which states that the application user first issues an SQL Insert, SQL Delete, OR SQL Update call **in the database** [thereby modifying the metadata in the database] and then links the file with certain constraints, which include, for example, making a database system the owner of the named file and marking the file as a read-only file. This linkage is provided in a transactional manner. The rationale for changing the owner of the file to the database system from a file system user is to prevent the file from being renamed or deleted by file system users, which guarantees the integrity of any reference made in the database system to the file. Marking the file as read only guarantees the integrity of indexes that may be created on the file and stored in the database system for search." This is quite opposite to the claimed invention in which the metadata and the file may be accessed without modifying the metadata and in which access to the metadata is permitted while the file/object is updated!

As discussed in the earlier response, a "transactional manner" to preserve data integrity, as in Burns '694, has a clear definition in the art of database management

with the following characteristics: a “transactional manner” is ATOMIC meaning that either all or none of the operations are completed; is CONSISTENT meaning that all transactions must leave the database in consistent state; is ISOLATED meaning that the transactions can’t interfere with each other’s work and incomplete work isn’t visible to other transactions; and is DURABLE meaning that successful transactions must persist through crashes of the object/file. Quite simply, because of this “transactional manner”, Burns ‘694 cannot suggest or teach that the object/file can be modified without modifying the metadata.

The Examiner then proposes that the version number, timestamp, operational priority, and node priority parameters as taught by Guturu ‘075 provide sufficient teaching and motivation to modify Burns ‘694. Applicant respectfully continue to traverse because neither Guturu ‘075 nor Burns ‘694 solve the stated problem of maintaining consistency between file content and the associated metadata, while permitting multiple updates to the file content. Guturu ‘075 solves the problem of synchronizing identical and redundant databases. Just as Burns ‘694 does not have the claimed elements of the amended independent claims, Guturu ‘075 also does not “update the externally stored object/file without modifying the metadata in the database.” Guturu ‘075 assumes a “tightly coupled” or a “transactional manner” in that the metadata related to the object being updated is never separated from the object. Both Burns ‘694 and Guturu ‘075 update the object and the metadata at once and do not suggest otherwise. Respectfully, comparing version numbers, timestamps and priorities does not allow updating the file/object without modification of the related metadata, as claimed by Applicant. Applicant further maintains that, as correctly noted by the Examiner, Burns ‘694 does not refer to an **encrypted access token**; the token **having a hash value**; the hash value **including a version number**; respectfully neither does Guturu ‘075. Applicant acknowledges that the Examiner will allow claims 1-11, 14-18, 20-30, 33-37, 39-48, and 51-55. The remaining rejected claims 19, 38, 56, 57, 58 are cancelled.

GROUP ART 2172
AMENDMENT AFTER FINAL Patent

Conclusion

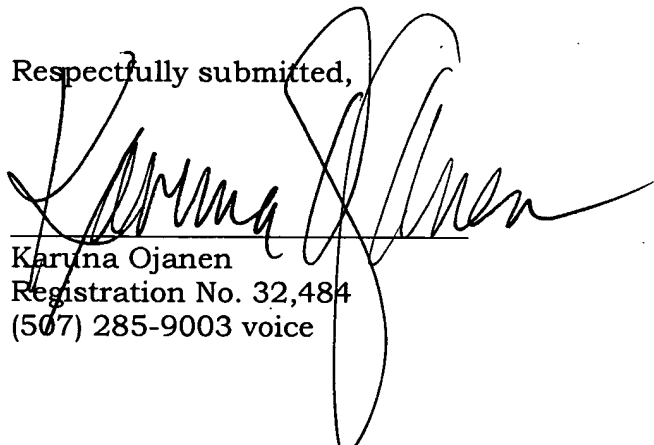
Applicant amends the independent claims to particularly point out and distinctly claim that the file/object can be updated without modification of the related metadata, which amendment is neither taught or suggested by Burns '694 nor by Guturu '075. Applicant amends claim 52 to correct a typographical error. Applicant requests the Examiner to review the amended claims, to enter the amendments because they incorporate the limitations of claims that have already been examined and that have indicated as being allowable. If the Examiner thinks any outstanding issues remain for issuance of the patent, he is encouraged to telephone the Attorney listed below.

Date: 18 October 2004

Ojanen Law Offices (OLO)
2665 Riverside Lane NE
Rochester, MN 55906-3456
(507) 252-5345 fax

Respectfully submitted,

By


Karuna Ojanen
Registration No. 32,484
(507) 285-9003 voice